Attestation in TLS
Introduction

• Historically Transport Layer Security (TLS) protocol has relied on Public Key Infrastructure (PKI) for authentication

• Remote Attestation presents an enhancement to PKI, leveraging hardware features to provide comprehensive information about the security state of the device

• Our work is focused on standardising remote attestation as a native authentication mechanism in TLS

• Also backed by an Open-Source proof of concept project, backed by Confidential Consortium Attestation Special Interest Group
TLS Extensions

• We’re defining a new certificate type
• Attestation metadata is carried instead of (or together with) an X.509 certificate
• Attestation metadata itself is opaque to the TLS implementation
• Also defining new TLS extensions that allow negotiation of the credential type, and conveyance of freshness
• Authenticating both Client and Server
Key & Platform attestation

Combined Attestation Bundle

Platform Attestation Token

Key Attestation Token

Attester

Workload

OS kernel

Firmware

Bootloader

Hardware (RoT)

Workload identity key

Key attestation service stack

Key Attesting Key
Client \rightarrow Server

ClientHello

\{…\}
evidence_proposal(types(a,b,c))

POST /newSession

201 Created
Location: /76389
Body: {
nonce,
Types(a,b,c,d,e)
}

ServerHello

\{…\}

Verifier \rightarrow Client

EncryptedExtensions

\{…\}
evidence_proposal ( nonce, 
type(a))
Message Flow

Attestation Service

Client

Server

Verifier

CertificateRequest
Certificate
CertificateVerify
Finished

Certificate\(\text{CMW}\{\text{mt, } \text{CAB}\}\)
CertificateVerify\(\text{sig}\)
Finished

\(\text{attest\_key} (\text{nonce, } \text{TIK})\)

\(\text{CAB (KAT, PAT)}\)

\(\text{sign (TIK, hs)}\)

\(\text{sig}\)
Message Flow

**Client**

**Attestation Service**

**Server**

POST /76389

Body :
CMW(mt, CAB)

Body :
att-result: AR{}

**Verifier**

verify AR{}

verify sig

application data
Mapping to RATS

Verifier

CoRIM

RV/EV provider

Supply Chain

Relying Party

CMW (Evidence)

EAR

Attester

CMW (Evidence)

TLS Handshake
(Certificate Message)
# Usage of RATS Draft

<table>
<thead>
<tr>
<th>Draft Name</th>
<th>Describes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attestation in TLS and DTLS</td>
<td>Describes TLS extensions to use attestation for authentication</td>
</tr>
<tr>
<td>EAT based Key attestation Token</td>
<td>Evidence format of combined key and platform attestation</td>
</tr>
<tr>
<td>CoRIM</td>
<td>Concise Reference Integrity Manifest (CoRIM), a standardised way to convey Reference Values and Endorsed Values to a Verifier</td>
</tr>
<tr>
<td>EAT Attestation Results (EAR)</td>
<td>An EAT profile for conveyance of Attestation Results</td>
</tr>
<tr>
<td>CMW</td>
<td>A format used to Wrap RATS Messages in a protocol agnostic way</td>
</tr>
<tr>
<td>EAT Collection Types</td>
<td>An extension to EAT allowing the top-level token to consist of a collection of otherwise defined tokens</td>
</tr>
</tbody>
</table>
Current Activities & Collaborations

Specifications

Formal Verification
TU Dresden

Implementation Prototype
Community Members

Specifications
Community Members
Prototype Architecture

Server
TLS Stack (MbedTLS)

Handshake (Augmented)

Attestation Evidence

Secure Channel

Verifier

Client
TLS Stack (MbedTLS)

Client Private + Attestation Key(s)

Platform State

Client RoT

Server Certificate

VERAISON

PARSEC
# Main Open-Source Repositories

<table>
<thead>
<tr>
<th>Repository Name(link)</th>
<th>Contains</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCC Attested TLS PoC</td>
<td>Central space for open collaboration on the proof of concept</td>
</tr>
<tr>
<td>Parsec</td>
<td>Library to abstract Attester Evidence Formats</td>
</tr>
<tr>
<td>Mbed TLS library</td>
<td>TLS Library</td>
</tr>
<tr>
<td>Veraisson</td>
<td>Attestation Verification deployment</td>
</tr>
<tr>
<td>ctoken</td>
<td>A C library to implement EAT, CWT and UCCS</td>
</tr>
<tr>
<td>t_cose</td>
<td>A C Library to implement COSE RFC 9052</td>
</tr>
</tbody>
</table>
Status of Implementation

• An end-to-end working proof of concept is now available
  • From Attester through a TLS implementation, to a Verifier
  • Uses Background check model, with TPM 2.0 as a RoT

• Open-Source availability of entire stack
  • The components themselves are open-source software
  • Project harbored under CCC-Attestation SIG

• Work In Progress on a Confidential Computing (CC) version of Attester running in a confidential environment (ARM-CCA) and performing an end-to-end Attested TLS handshake between Client (Attester), Server (RP) and a Verifier

• Community Members Welcome to engage in implementation to bring in other attestation formats
Future Work

• To enhance draft to include RATS passport model
• Implement a PoC based on the passport model